Nolen Young

11517296

Big Data HW1

1. In social media, all data possible is tracked from the user. The data is used to understand what the user finds entertaining and better tailor the content presented to the user in order to try and keep them on the site for as long as possible. They track things like likes, comments, links that you click on, and even how long you look at a post. They also try to tag and categorize user generated content so that they can present that user generated content to the people who will find it most interesting. The velocity of this data is very high. Data is being generated by all users on the platform constantly, and all that data needs to be analyzed and stored very quickly to match the rate of the data being generated by users. This also means the volume of data is very high, depending on the size of the social network. A lot of social media sites have hundreds of millions of users, and with each user generating constant data, a lot of data is collected. The data is also very valuable. The data allows social media sites to make the content presented to each user personalized to that persons tastes and interests. Having a social media site personalized makes users use the site more often, and stay on the site longer, allowing more valuable data to be collected and more ads to be presented to the user for profit. This data is completely unstructured, and contains videos, pictures, likes, comments, and all other user generated data. Because of this it would be best to use and RDF graph to represent the data. The most important thing is establishing connections between content and users, and a graph is the best way to represent connections between unstructured data. The veracity of the data is very low, however, because all the data is user generated. Any user generated content is bound to be inaccurate due to the inaccuracy of humans themselves.
   1. The relations scheme is the set of all the attributes in the table. The relational database schema includes the airport ID as a primary key for the table. The domain of the table is all airports, train stations, and ports. The attributes are the categories of data stored in the table, ie name, city, country, latitude, longitude etc. The attribute domain is the set of all possible values of each attribute. So the country attribute has the domain of all country names. A relation instance is the filled-out table with data, or no data, that is valid to the relation schema. Ex: (1, Seatac, Seattle, United States, null, null, -32, 5, 20, 7, America/Los\_Angeles, airport, u) (2, Spokane International, Spokane, United States, 123, 4321, -32, 5, 20, 7, America/Los\_Angeles, airport, u) (3, LAX, Los Angelas, United States, 321, 1234, -32, 3, 20, 7, America/Los\_Angeles, airport, u) (4, Portland International, Portland, United States, 132, 1423, -32, 4, 20, 7, America/Los\_Angeles, airport, u)
   2. Airport

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. ID | Name | City | Country | IATA | ICAO | Latitude | Longitude | Altitude | Timezone | DST |
| Tz | Type | Source |

Airline

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Alias | IATA | ICAO | Callsign | Country | Active |

Route

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Airline | *Airline ID* | Source Airport | *Source ID* | Destination Airport | *Destination ID* | Codeshare | Stops | Equipment |

c.

Augmentation Rule: City -> Country, so City, Timezone, -> Country, Timezone Transitivity rule: Alias -> Name, name -> callsign, so Alias -> callsign

a.

1. PiTitle(SelectionTitle=Zootopia(Schedule))

PiTheater, Address (SelectionDirector = Steven Spielburg(Movies Theta JoinTitle = Title (Location Theta JoinTheater = Theater Schedule))

PiAddress, Phone Number(SelectionTheater = Le Champo Theater(Location)

PiActor, Actor2(Movies Theta JoinTitle = Title2, Director = Director2, Actor != Actor2 p(Movies -> Movies2, Title -> Title2, Director -> Director2, Actor -> Actor2))

<Airport id = 1>

<Name> Seatac </Name>

<City> Seattle </City>

<Country> USA </Country>

<IATA> null </IATA>

<ICAO> null </ICAO>

<Latitude> -32 </Latitude>

<Longitude> 5 </Longitude>

<Altitude> 20 </Altitude>

<Timezone> 7 <Timezone>

<DST> A </DST>

<Tz> America/Los\_Angeles </Tz>

<Type> airport </Type>

<Source> u </Source>

</Airport>

<Airport id = 2>

<Name> Spokane International </Name>

<City> Spokane </City>

<Country> USA </Country>

<IATA> 123 </IATA>

<ICAO> 4321 </ICAO>

<Latitude> -32 </Latitude>

<Longitude> 5 </Longitude>

<Altitude> 20 </Altitude>

<Timezone> 7 <Timezone>

<DST> A </DST>

<Tz> America/Los\_Angeles </Tz>

<Type> airport </Type>

<Source> u </Source>

</Airport>

<Airport id = 3>

<Name> LAX </Name>

<City> Los Angeles </City>

<Country> USA </Country>

<IATA> 321 </IATA>

<ICAO> 1234 </ICAO>

<Latitude> -32 </Latitude>

<Longitude> 3 </Longitude>

<Altitude> 20 </Altitude>

<Timezone> 7 <Timezone>

<DST> A </DST>

<Tz> America/Los\_Angeles </Tz>

<Type> airport </Type>

<Source> u </Source>

</Airport>

<Airport id = 4>

<Name> Portland International </Name>

<City> Portland </City>

<Country> USA </Country>

<IATA> 132 </IATA>

<ICAO> 1423 </ICAO>

<Latitude> -32 </Latitude>

<Longitude> 4 </Longitude>

<Altitude> 20 </Altitude>

<Timezone> 7 <Timezone>

<DST> A </DST>

<Tz> America/Los\_Angeles </Tz>

<Type> airport </Type>

<Source> u </Source>

</Airport>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="Airport">

<xs:complexType>

<xs:sequence>

<xs:element name = "ID" type = "xs:int"/>

<xs:element name = "Name" type = "xs:string"/>

<xs:element name = "City" type = "xs:string"/>

<xs:element name = "Country" type = "xs:string"/>

<xs:element name = "IATA" type = "xs:int"/>

<xs:element name = "ICAO" type = "xs:int"/>

<xs:element name = "Latitude" type = "xs:float"/>

<xs:element name = "Longitude" type = "xs:float"/>

<xs:element name = "Altitude" type = "xs:int"/>

<xs:element name = "Timezone" type = "xs:int"/>

<xs:element name = "DST" type = "xs:char"/>

<xs:element name = "tz" type = "xs:int"/>

<xs:element name = "type" type = "xs:string"/>

<xs:element name = "source" type = "xs:char"/>

</xs:sequence>

</xs:complexType>

<xs:key name="PrimaryKey">

<xs:selector xpath="Airport"/>

<xs:field xpath="@ID"/>

</xs:key>

</xs:element name="Airport>

<xs:element name="Airline">

<xs:complexType>

<xs:sequence>

<xs:element name = "ID" type = "xs:int"/>

<xs:element name = "Name" type = "xs:string"/>

<xs:element name = "Alias" type = "xs:string"/>

<xs:element name = "Country" type = "xs:string"/>

<xs:element name = "IATA" type = "xs:int"/>

<xs:element name = "ICAO" type = "xs:int"/>

<xs:element name = "Callsign" type = "xs:string"/>

<xs:element name = "Active" type = "xs:char"/>

</xs:sequence>

</xs:complexType>

<xs:key name="PrimaryKey">

<xs:selector xpath="Airline"/>

<xs:field xpath="@ID"/>

</xs:key>

</xs:element name="Airport>

<xs:element name="Route">

<xs:complexType>

<xs:sequence>

<xs:element name = "Airline" type = "xs:string"/>

<xs:element name = "Airline ID" type = "xs:int"/>

<xs:element name = "Source Airport" type = "xs:string"/>

<xs:element name = "Source Airport ID" type = "xs:int"/>

<xs:element name = "Destination Airport" type = "xs:string"/>

<xs:element name = "Destination Airport ID" type = "xs:int"/>

<xs:element name = "Codeshare" type = "xs:char"/>

<xs:element name = "Stops" type = "xs:string"/>

<xs:element name = "Equipment" type = "xs:int"/>

</xs:sequence>

</xs:complexType>

<xs:key name="PrimaryKey">

<xs:selector xpath="Route"/>

<xs:field xpath="@Airline"/>

<xs:field xpath="@Source Airport"/>

<xs:field xpath="@Destination Airport"/>

</xs:key>

<xs:keyref name = "Airline Key">

<xs:selector xpath="Route"/>

<xs:field xpath="@Airline ID"/>

<xs:keyref name = "Airline Key"/>

<xs:keyref name = "Source Key">

<xs:selector xpath="Route"/>

<xs:field xpath="@Source Airport ID"/>

<xs:keyref name = "Source Key"/>

<xs:keyref name = "Destination Key">

<xs:selector xpath="Route"/>

<xs:field xpath="@Destination Airport ID"/>

<xs:keyref name = "Destination Key"/>

</xs:element name="Airport>

I encode primary keys using the key tag and set the path to that key. I do the same thing with foreign keys except I use the keyref tag instead.

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:si="https://www.w3schools.com/rdf/">

<rdf:description id = "human">

<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

<rdf:property id="gender" type = "xs:string"/>

<rdf:property id = "BirthYear" type = "xs:year"/>

</rdf:description>

<rdf:description id = "man">

<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

<rdf:subClassOf rdf:recourse = "#human"/>

<rdf:property id ="gender" type = "xs:string">man</rdf:property>

<rdf:property =id "isFatherOf" rdf:recourse = "#human"/>

<rdf:property = id "isParent" rdf:recourse = "#isFatherOf"/>

</rdf:description>

<rdf:description id = "woman">

<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

<rdf:subClassOf rdf:recourse = "#human"/>

<rdf:property id ="gender" type = "xs:string">woman</rdf:property>

<rdf:property id = "isMotherOf" rdf:recourse = "#human"/>

<rdf:property = id "isParent" rdf:recourse = "#isMotherOf"/>

</rdf:description>